A PERFORMANCE MODEL AND SCALABILITY ANALYSIS OF THE HYCOM OCEAN SIMULATION APPLICATION

Kevin J. Barker and Darren J. Kerbyson
Performance and Architecture Laboratory (PAL)
CCS-3, Computer and Computational Sciences Division
Los Alamos National Laboratory, USA
{kjbarker,djk}@lanl.gov

ABSTRACT
In this work we present a detailed analytical performance model of the large-scale parallel application HYCOM, the Hybrid Coordinate Ocean Model. The performance model is developed from analyzing the activities contained within the code. It is parameterized in terms of both application characteristics (including problem sizes, time-steps, and application phases) and system characteristics (including single processor performance, computational node size, network latency, and network bandwidth). Through a validation stage it is shown that the model is reasonably accurate with a typical prediction error of 10% when compared with actual run-times across three parallel systems and when using three different input decks. Once developed and validated the model may be used to explore performance scenarios, consisting of application and system combinations, for which it may not be possible to directly measure. One study is presented here which is used to provide insight into the achievable performance on a possible future (2010 timeframe) system. In addition we use the performance model to show the sensitivity of HYCOM's run-time to the performance of the individual sub-systems: processor speed, network latency, and network bandwidth. These studies illustrate the power of performance modeling once an accurate model has been developed.

KEY WORDS
Performance modeling, performance analysis, and large-scale applications

1. Introduction

There is, as ever, a need to increase the performance of high-end computing systems so as to both decrease the time-to-solution of current applications whilst also making possible increased resolutions in the physical simulations and increased complexity of the physics that are utilized. Current and future needs of science enabling simulations have been recently reviewed for the Department of Energy [1] which illustrate the need for much improved computational resources. In addition, programs including the DARPA High Productivity Computing Systems (HPCS) are looking to develop multi-Petaflop scale systems in the 2010 time-frame. Japan and China have also recently announced similar goals for high end systems. A follow-on to the current highest-peak system, the Blue Gene/L system at Lawrence Livermore National Laboratory, may also have a petaflop peak performance prior to this time.

However, as is repeatedly stated, the peak performance of a system is not the prime interest when assessing the performance of computing systems – rather the most important metric is the run-time of the workload that the system is used to process. This is the only real metric that is useful, but it is application dependent – one machine does not give the highest performance for all applications. The run-time can then be used to translate into other important application-specific measures such as how long a given problem will take, or what size problem can be processed given a large-scale system of a given size and given time allocation.

Climate modeling is one of many important scientific domains that have been identified as an increasingly important area for high performance computing whose computational requirements may way well grow by up to 10 orders of magnitude [1]. There is a need to increase the current spatial resolution being processed, to add new physics/science models, to increase the physical time being simulated, and to perform multiple ensembles to assess model variability.

In this work we analyze the performance of the Hybrid Coordinate Ocean Model (HYCOM) that was originally developed jointly between the University of Miami, Los Alamos National Laboratory (LANL), and the Naval Research Laboratory (NRL) [2]. The goal of this work is to develop a detailed analytic performance model which has the capability to accurately predict the performance of the application (i.e., execution time) for various large-scale systems at different machine scales (number of processing elements) and for a variety of input problem sizes and physical models. Such a performance model can be used to ascertain the expected performance of an application on various existing systems as well as the performance on possible future computing platforms after it has undergone a validation – that is the performance model accurately matches measurements made on current systems. It may also be used to gain insight into the relative importance of various machine characteristics, such as processing speed, network latency, and network bandwidth.

The process that we use to develop a performance model is application centric – we develop an understanding of the activities within the application, the data decomposition methods that are utilized, the way in which individual tasks are mapped across the system, and the processing flow that occurs within an iteration of the application. This leads to the costing of the application activities that also take into account typical contention arising from both within the computational node and within the network communication fabric for inter-node communications. Inputs to the model include: application independent characteristics of the system including communication network latency and bandwidths; application run-time on a single-processor of the target system; and contention that arises within a single computation node. In this way, measurements from a small scale system (typically two nodes) are used as input to the model that can be utilized to predict the performance of a large-scale system. If a single node is not available for measurement such as the case for a future system, single-node performance simulations (or other data) can be used in replacement.

We present a detailed performance model of HYCOM in this work that is applicable across machines and for a wide-variety of problem inputs. The model is validated against measurements on three systems available to us up to system of size 512 processors. The model is shown to have a high prediction accuracy, to within an average of 10% of actual measurement. The availability of this model will enable a wide-range of performance studies to be carried out not only in terms of altering the problem size inputs to the application and observing the achievable performance changes, but also in terms of altering the characteristics of the system and hypothesizing the make-up of possible future systems. In this work we consider the achievable performance of HYCOM on a future system which contains up to 256K processors operating at 8GHz and having an interconnection bandwidth of 8GB/s and latency of 2μs. This illustrates the real purpose for developing the performance model in the first place.

This work is part of an on-going effort to model the performance of a wide range of applications on existing and possible future large-scale systems. Applications already modeled include: radiation transport, hydro-codes, and of most relevance to this work – the Parallel Ocean Program (POP) [3, 4] which implements different physics in simulating the ocean flow. These models have been used in many activities including: the comparison of large-scale systems [5], the optimization of ASCI Q at Los Alamos [6], and also in system procurements.

In Section 2 an overview of HYCOM is provided in terms of its input, the decomposition of the spatial grid and the typical operations that it performs. In Section 3 we detail the performance model constructed from the activities contained within an iteration of HYCOM. The validation in Section 4 shows that the resultant model has a high predictive capability. The model is then used to analyze the performance of a possible future system whilst also looking at the sensitivity of sub-system performance on the overall run-time of HYCOM.

2. Overview of HYCOM

HYCOM is a general ocean circulation model that employs a hybrid coordinate scheme to improve modeling accuracy as the ocean depth varies from deep stratified water to shallow coastal regions [2]. As such it represents an improvement over earlier software developed at the University of Miami; vertical coordinates remain isopycnic in the open ocean and transition smoothly to z-coordinates in the weakly stratified open ocean mixed layer, to sigma coordinates in shallow water regions, and back to z-coordinates in very shallow regions. Currently, HYCOM is managed by the multi-institutional HYCOM consortium, funded by the National Oceanic Partnership Program (NOPP) as part of the United States Global Ocean Data Assimilation Experiment (GODAE) for the purpose of developing a 3-dimensional representation of the ocean state at a fine-grained resolution in real-time and studying ocean-land and ocean-atmosphere boundary conditions [7]. HYCOM has been used to study the effects of intra-seasonal atmospheric variability on surface current in the equatorial Indian Ocean region and to conduct 20 year simulations of the tropical Pacific and Atlantic Oceans, to name just a few [8, 9].

HYCOM version 2.1 is composed of approximately 25 thousand lines of code written in Fortran 90. Source code for HYCOM is available from the HYCOM consortium [10] It is portable to all Unix-based systems, and is fully parallel, utilizing either shared memory (in the form of OpenMP) or MPI message passing for communication between processing elements in a distributed memory system. In this work, we focus strictly on the distributed memory implementation using MPI.

Three data sets are used to analyze the performance of HYCOM in this work denoted as small, medium and large (listed in Table 1). They differ in the spatial grid resolution and in the physical ocean region they represent. The medium and large cases represent the whole world at different resolutions. The small case represents a part of the Pacific Ocean.

2.1 Data Decomposition

The input domain to HYCOM consists of a 3-dimensional (*i*, *j*, and *depth*) spatial grid that represents ocean or land regions. This input domain is subdivided into tiles by first partitioning the *j* dimension of the overall spatial grid, creating a series of rows of constant height.

Each row is then subdivided along its *i*-dimension, creating a total of *T* tiles each having a constant number of vertical layers as defined by *depth*. Although each row is of equal height, each tile within a row may be of

arbitrary width, resulting in differing numbers of varying sized tiles in each row. This partitioning scheme, impacts the computation required per tile and also the size of the boundary interfaces between tiles which in turn impacts the communication requirements in a parallel system.

Table 1: Input decks used to analyze HYCOM

Input-deck	Oceans	Spatial grid	Resolution
Small	Pacific	450×450×22	1/12 deg.
Medium	World	1500×1100×26	1/4 deg.
Large	World	4500×3298×26	1/12 deg.

Any tiles that consist of just a region of land are removed from further consideration, reducing the number of tiles that will be processed. Only one tile is assigned to any processor, thus $T_{ocean} = T - T_{land}$ processors are required where T_{land} is the number of land-only tiles.

Figure 1 depicts a map of the world in which land regions are shown in black, and oceans are shown in color. Individual tiles are denoted by different colors for two cases. The first case (Figure 1, top) consists of 504 tiles, and the second case consists of 5107 tiles. North America can be seen in the upper-middle part of both diagrams and is more readily discernible when using a greater number of tiles. Each tile is mapped onto a separate processor. As the number of processors increases, the amount of computation assigned to a single processor is reduced. This is typical of a *strong* scaling mode in which the global problem size is a constant and parallelism is used to decrease the time-to-solution. HYCOM's input domain may be toroidal in the *i* (horizontal) dimension only, as in the case shown in Figure 1, in which the input domain wraps-around the globe.

2.2 Scaling Characteristics

As HYCOM is typically run in *strong* scaling mode, the average tile size is dependent on the number of available processors. Assuming *P* processors and an input domain of size $i \times j \times depth$, the average volume is simply the global number of spatial grid points divided by *P* and the boundary sizes between tiles decreases by the square root of *P* in each dimension. The tile volume and tile boundary sizes are listed in Table 2. Note that for simplicity we assume that the spatial grid does not include land regions. If land were included, then the values listed in Table 2 would be lower.

It can be seen that, in *strong* scaling mode, the tile volume decreases more rapidly than the boundary surface area as the number of processors increases. Therefore, as the processor count scales, the performance of HYCOM will become bound by communication costs. Since the tile boundaries also decrease in size with processor count, the message cost will be increasingly sensitive to the communication latency in a parallel system.

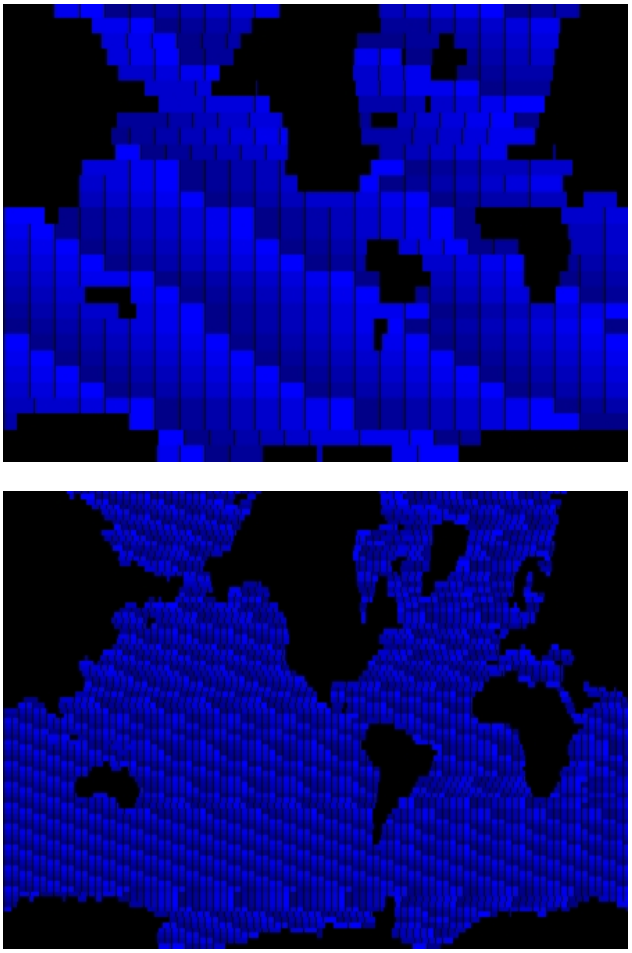


Figure 1: Example partitioning of the world's oceans into 504 (top) and 5107 (bottom) tiles; blue tiles indicate ocean while black indicates regions of land

2.3 Processing flow

The computation in HYCOM is composed of two primary phases known as *baroclinic* and *barotropic*. However, the baroclinic phase requires the most computation cycles. The simulation is composed of a number of cycles, each of which has one baroclinic phase that contains an even number of barotropic phases. Each baroclinic phase consists of several steps which are listed in Table 3. Each

Table 2: Tile boundary area and volume in terms of input dimensions and number of processors

Tile volume	Tile boundary *i*	Tile boundary *j*
$V = \frac{i \times j}{N_{PE}} depth$	$B_i = \frac{j}{\sqrt{N_{PE}}} depth$	$B_j = \frac{i}{\sqrt{N_{PE}}} depth$

step of the baroclinic phase contains both computation and communication components. In order to simplify the presentation of the performance model in Section 3, we aggregate all computation components into a single term which is parameterized by the number of barotropic steps per baroclinic phase and tile volume.

Table 3: HYCOM baroclinic stages

Step	Description
Cnuity	Flux-corrected continuity calculation.
Tsadvc	Advection and diffusion calculation
Momtum	Hydrostatic calculation
Barotp	Barotropic calculation
Thermf	Thermal forcing functions
Hybgen	Hybrid coordinate grid generator
Mxkprf	Vertical diffusion and mixing models

The communication between processors takes two forms. The most frequent are boundary exchanges, in which processors update a pre-defined "halo" region on neighboring processors, i.e. for those tiles that share a boundary interface. The second type is a software reduction, in which processors send messages to a single target, either at the head of the processor's row or the head of the first processor column.

a) Boundary exchanges

Figure 2 depicts the communications that are required in order for the center tile to update its immediate neighbors. In the case of a regular layout in which every row is evenly divided into tiles (Figure 2, upper-left, which is equivalent to the case when no land regions are included in the spatial grid), a tile may have at most four immediate neighbors. However, because rows may be arbitrarily divided into tiles, an irregular tile layout (Figure 2, upper-right) often results. In this case, a tile may have more than one neighbor to the North or South, resulting in a greater number of messages required for a boundary update. However, the sizes of such messages are scaled by the size of the interface between tiles.

A third possibility occurs when a tile containing ocean neighbors a region of land. In this case, a gap in the domain results as the land tiles are removed from the computation (Figure 2, bottom). Neighboring relationships do not span such gaps; therefore no boundary update message is required. Note that the boundary updates are bi-directional – i.e. the center tile in Figure 2 not only sends boundary data to its neighbors but also receives data from them.

b) Software reductions

The second type of communication is a software "reduction" operation, different from a collective type operation, in which multiple processors communicate with a single target node. There are two steps in this reduction: 1) all processors in a row send a message to the processor at the head of that row (termed *row-reduction*), and 2) the head of each row sends a message to the "root" processor (termed *column-reduction*).

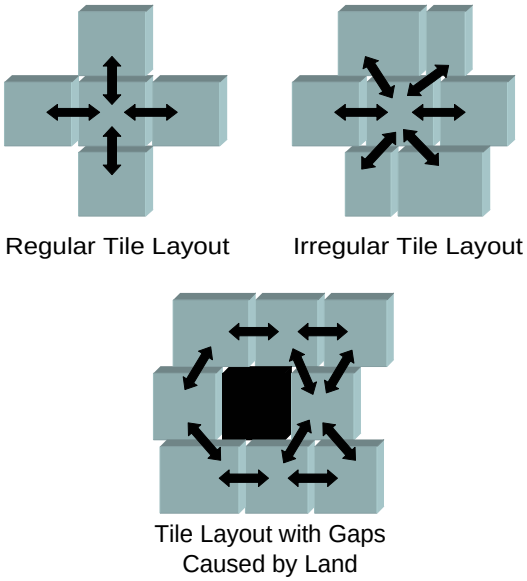


Figure 2: Boundary update communication patterns

The size of each message is dependent upon the dimensions of the tile on the processor sending the message. However, the number of messages per baroclinic cycle is largely independent of the specific problem instance. The one exception is the messages that are sent within the barotropic phase; the number of these messages varies linearly with the number of barotropic phases per baroclinic phase.

3. HYCOM Performance Model

Computation within a single iteration is defined by the seven steps described in Section 2.3. Each step consists of computation and inter-processor communication. In order to simplify the model's presentation, we aggregate the computational components contained within these steps into a single compute term. Therefore, the time required for a single iteration is composed of the computational steps (excluding the barotropic), the time required for barotropic phases, the time incurred by contention within a multi-processor node (consisting of both memory contention, operating system overhead, and intra-node communication contention), and inter-processor communication.

HYCOM's run-time is modeled as follows:

$$T_{cycle}(N_{PE}, C_{PE}, N_{BT}, N_{SMP}) = T_{comp}(C_{PE}) + N_{BT}T_{BT}(C_{PE}) + T_{memory}(N_{SMP}, C_{PE}) + T_{comm}(N_{SMP}, C_{PE})$$

The HYCOM performance model parameters are listed in Table 4. The first set of parameters describes the problem spatial grid sizes, the second set describes the system configuration as well as application independent network communication times, and the third set denotes the measurements taken from a single processor execution.

Table 4: HYCOM input parameters

Parameter	Description	Measured Derived or Input
I	Horizontal dimension	Input
J	Vertical dimension	Input
Depth	Layers in ocean	Input
C_{PE}	Tile Volume	Derived
N_{BC}	Total number of baroclinic phases	Input
N_{BT}	Number of barotropic phases per baroclinic	Input
N_{PE}	Processor count	Input
N_{SMP}	Number of processors in an SMP node	Input
$T_{comp}(C_{PE})$	Computation time excluding barotropic	Measured
$T_{BT}(C_{PE})$	Barotropic computation time	Measured
$T_{memory}(N_{SMP}, C_{PE})$	Time due to contention within SMP node	Measured
$L(S)$	MPI network latency	Measured
$T_B(S)$	MPI network bandwidth	Measured
$T_{comm}(N_{SMP}, C_{PE})$	Inter-processor communication time	Derived

3.1 Inter-processor Communication

Communication between processors consists of either point-to-point (a message sent from a source processor to a single target processor) or collective (involving groups of processors). Basic point-to-point message passing time is modeled by the following piece-wise linear equation:

$$T_{msg}(S) = L(S) + S.T_B(S)$$

where $L(S)$ is the effective latency and $T_B(S)$ is the effective time-per-byte of a message of size S Bytes. Both the network latency and bandwidth are measured on the target system for bi-directional communication. Collectives are modeled by using a binary "fan-in, fan-out" message passing strategy in a binary tree structure. Such a method requires a total of $2\log_2 P$ messages.

Contention within the system network from multiple messages using the same communication channels at the same time will affect communication performance, and can arise when multiple processors (such as those with a single SMP) contend for limited network resources. The basic message passing model is expanded to describe contention in Section 3.2.

Table 5: Summary of HYCOM communication per processor per iteration by step

Step	Operation	Count	Size
Cnuity	East/West	6	(T_j+2*Halo*)**depth***Halo*
		6	(T_j +2*Halo*)**Halo*
	North/South	6	T_i **depth***Halo*
		6	T_i **Halo*
	Collectives	1xAllReduce	*depth**8
		1xAllReduce	*depth**16
Tsadvc	East/West	7	(T_j +2)**depth**2
	North/South	7	T_i **depth**2
	Reductions	4*depth* (row)	20* T_j
		4*depth* (col)	T_j
	Collectives	2xAllReduce	*depth**2
		4xBroadcast	1
Momtum	East/West	13	(T_j +2*Halo*)**Halo*
		13	(T_j +2*Halo*)*3*Halo*
		26	(T_j+2*Halo*)**depth***Halo*
	North/South	13	T_i **Halo*
		13	T_i *3*Halo*
		26	T_i **depth***Halo*
Barotp	East/West	2+(1.5*N_{BT})	(T_j +2*Halo*)**Halo*
		2+(1.5*N_{BT})	(T_j +2*Halo*)*3*Halo*
	North/South	2+(1.5*N_{BT})	T_i **Halo*
		2+(1.5*N_{BT})	T_i *3*Halo*
Thermf	Collectives	1xBarrier	1
		1xBroadcast	1
	Reductions	8.N_{BC} (row)	20 T_j
		8N_{BC}(column)	T_j
Hybgen	East/West	2	(T_j +2)**depth*
		2	T_j +2
	North/South	2	T_i **depth*
		2	T_i
Mxkprf	East/West	3	(T_j +2)**depth*
		3	T_j+2
	North/South	3	T_i**depth*
		3	T_i

Message traffic is categorized in Table 5 by application step and payload size. Message payload sizes are parameterized by tile dimensions (*i*, *j*, and *depth*), halo region size (the border depth of each tile that is updated in a neighbor exchange, *Halo*), the number of baroclinic steps (N_{BC}), and the number of barotropic phases per baroclinic (N_{BT}).

3.2 Multi-processor Induced Contention

From our experience of modeling many different systems, the main contention that arises in a parallel system is due to multiple processors within a node competing for a

single memory bus and for a single network interface. Memory contention is measured by noting the performance degradation as multiple single-processor copies of HYCOM are simultaneously executed on all processors within a single SMP node. Performance degradation is a function of the resulting pressure on the local memory system as well as any overhead induced by the operating system, which is a factor of the number of processors within a node being utilized as well as the volume of the local tiles.

Communication contention is a function of the number of simultaneous "out-of-node" messages. The majority of communication is in the form of boundary exchanges with either a processor's east/west or north/south neighbors as listed in Table 5. In the first case, the east-west boundary communications, the number of out-of-node messages is limited to two as shown in Figure 3(a), while in the second case, north-south boundary communications, the number rises to eight as shown in Figure 3(b). This contention factor is therefore the number of out-of-node messages divided by the number of available communication links and acts as a scalar multiple increasing the amount of time required for the message passing bandwidth component. For an SMP node, the number of communication links is typically one. Note also that, with some networks, the communication latency is also affected by contention. In this work we assume that the latency component of message passing time is mainly due to time spent in the messaging software stack local to the source and destination processors and therefore not subject to contention.

3.3 HYCOM Communication Cost

Collective communication operations are described in Section 3.1. Boundary exchange communication time, taking into consideration the effects of network contention discussed in Section 3.2, is given by:

$$T_{bound}(N_{SMP}, C_{PE}) = 2\sum_{i=steps}\sum_{j=comm_i} N_{EW_{i,j}} T_{msg}(S_{EW_{i,j}}) + 2\frac{N_{SMP}}{C_L}\sum_{i=steps}\sum_{j=comm_i} N_{NS_{i,j}} T_{msg}(S_{NS_{i,j}})$$

The two terms in this equation correspond to the east-west and north-south communications respectively as illustrated in Figure 3. The summations are taken over the steps listed in Table 5, with $N_{EWi,j}$ and $N_{NSi,j}$ being the number of east-west and north-south messages in step i, and the respective sizes being indicated by $S_{EWi,j}$ and $S_{NSi,j}$, all of which are listed in Table 5. Each step contains $comm_i$ communications of different sizes. Note that some of the steps have zero boundary communications.

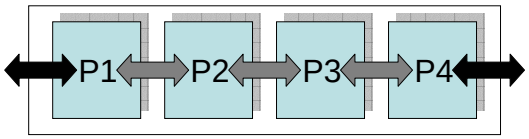
Row and Column reductions are modeled as:

$$T_{reduction}(C_{PE}) = \sum_{i=steps}\left(\sqrt{N_{PE}}\cdot T_{msg}(S_{row_i}) + \sqrt{N_{PE}}\cdot T_{msg}(S_{col_i})\right)$$

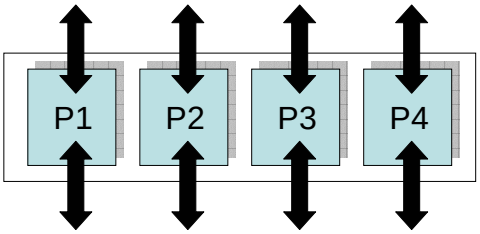
The sizes of the messages, S_{rowi} and S_{coli}, are again listed in Table 5. Note that, in this case, we are assuming the available processors are arranged in a "square" grid of dimension $\sqrt{N_{PE}} \times \sqrt{N_{PE}}$.

4. Model Validation

We have validated the HYCOM model on three input domain resolutions (Table 1: "Small", "Medium", and "Large") on three parallel machines. The first machine consists of quad-processor Hewlett-Packard/Compaq Alpha EV-68 nodes running at a clock speed of 833 MHz. Each node is equipped with 8 Gbytes of memory. The second platform is composed of dual-processor Intel Itanium nodes with processors running at a clock speed of 1.3 GHz. Each node contains 2 Gbytes of memory. The final machine contains 256 Hewlett-Packard/Compaq nodes, each consisting of four EV-68CD Alpha processors at a clock speed of 1.25 GHz.



(a) East/west communication



(b) North/south communication

Figure 3: Contention varies with the number of simultaneous sends competing for the communication links available to a node (Example with N_{SMP} = 4).

Each node is equipped with 16 Gbytes of memory. In each machine, each node is connected to a Quadrics QsNet fat-tree network via a single NIC. The characteristics and input parameters to the HYCOM performance model for these three machines are summarized in Table 6.

All parameters listed in Table 7, with the exception of the nodes used during validation, are measured through experimentation. Computation time per "cell" (a "cell" is defined as a single spatial grid point) is determined through single-processor runs on spatial grids of various resolutions. Network latency and bandwidth figures are derived from the results of a bi-directional "ping-pong" benchmark.

Table 6: Summary of the prediction errors of the HYCOM performance model

System	Input	Mean Error (%)
Alpha EV68	Small	17
	Medium	12
Itanium	Small	5.6
	Medium	8.5
Alpha EV68-CD	Medium	7.7
	Large	7.9

We model the runtime on the slowest processor in the system, i.e. the processor with the largest tile volume and largest communications costs. We therefore implicitly ignore the existence of gaps in the input spatial domain caused by regions of land; at present our model does not explicitly take land regions into account when predicting overall runtime.

Figure 4 depicts a comparison between run-time measurements and predictions for HYCOM across the multiple input-decks and systems. It can be seen that in each case the model is in good agreement with measured runtimes. With the exception of a couple of outlying points, our overall runtime predictions are within 10% of measured data as tabulated in Table 6.

5. HYCOM Scalability Analysis

With a validated performance model in place, we are able to examine application performance as input problem size and machine architecture scale. This allows us to quantitatively examine how HYCOM will behave on problems and machines that may be available in the near future and thus provide early input to the application users on what may be achievable in terms of problems that can be processed in a given time constraint. For applications such as HYCOM that execute in strong scaling mode, increases in future machine capability not only allow for the more rapid solution of current problems, but allow for the solution of larger problems on spatial grids with a greater degree of resolution.

In this Section, we examine two problem sizes: the "Medium" and "Large" domains specified in Table 1. The first problem represents a domain that are routinely soluble by today's standards, while the second is a spatial grid that we feel will be feasibly processed within a five year timeframe. In addition, we specify a possible future (again, approximately a five year timeframe) machine architecture, with processors operating at a speed of 8 GHz and with a fat-tree network with a latency of 2 µs and a maximum bandwidth per channel of 8 Gbytes/s. In this analysis, we scale the machine size up to 256K processors, and we assume that the computational efficiency of a node will be the same as that is achieved on the current systems.

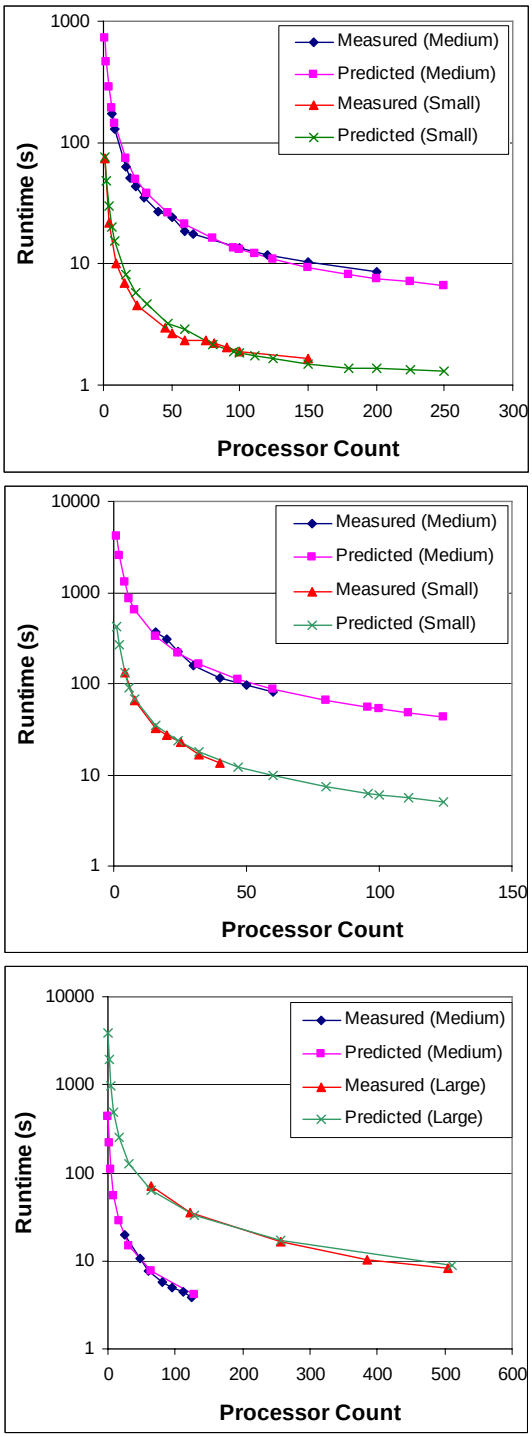


Figure 4: Model validation on HP Alpha EV-68 (top), Intel Itanium (center), and HP Alpha ES-68CD (bottom) platforms

In Figure 5, we examine the rate at which a HYCOM simulation of a particular duration can be completed expressed in the application specific metric of simulation years completed per day of system run-time. We assume each baroclinic time-step simulates 225 seconds (N_{BC} = 384 per day), and there are only two barotropic steps per

Table 7: System parameters used for model validation

Processor	HP Alpha EV-68	Intel Itanium	HP Alpha ES-45
Speed (MHz)	833	1300	1250
PEs/Node	4	2	4
# Nodes Used	50	30	126
Links/Node	1	1	1
Compute/Cell (µs)	21.096	110.026	10.070
Network Latency (µs)	*In-box* 14 0 – 64 12 64 – 512 35 512 – 4*K* 25 ≥ 4*K* *Out-of-box* 11 0 – 64 12 64 – 2*K* 0 ≥ 2*K*	*In-box* 11 0 – 32 12 32 – 512 36 512 – 8*K* 6 ≥ 8*K* *Out-of-box* 7 0 – 32 8 32 – 1*K* 15 ≥ 1*K*	*In-box* 9 0 – 32 8 32 – 512 32 512 – 8*K* 47 ≥ 8*K* *Out-of-box* 7 0 – 32 6 32 – 512 14 ≥ 512
Network Bandwidth (Mbyte/s)	*In-box* 26 0 – 64 22 64 – 512 5*E*3 512 – 4*K* 294 ≥ 4*K* *Out-of-box* 50 0 – 32 64 32 – 1*K* 78 ≥ 1*K*	*In-box* 1*E*3 0 – 32 21 32 – 512 5*E*3 512 – 8*K* 435 ≥ 8*K* *Out-of-box* 63 0 – 32 56 32 – 1*K* 91 ≥ 1*K*	*In-box* 278 0 – 32 22 32 – 512 2*E*3 512 – 8*K* 455 ≥ 8*K* *Out-of-box* 115 0 – 32 48 32 – 512 132 ≥ 512
Memory Contention (µs/Cell/PE)	10.5	23.9	17.1

baroclinic (N_{BT} = 2). We can see that in order to complete a single simulation year within one day of run-time on the 1/4 degree (medium) resolution spatial grid, roughly 128 processors are required. However, to complete the same simulation within the same time constraint on a spatial grid with 1/12 degree resolution, 1024 processors are required. Further, we can see that the simulation on the coarser resolution grid stops scaling at roughly 4096 processors; at this point tile volume is small enough that no benefit is derived from further parallelization and the penalties associated with increased communication come to dominate the run-time. Simulations on the spatial grid resolved to 1/12 of a degree continue to scale up to 16K processors before communication costs prohibit further improvements in execution time.

Figure 6 explores the sensitivity of HYCOM's overall run-time to the processing speed, network latency and network bandwidth on the spatial grid of 1/12 degree resolution. By varying the performance of each sub-system from the given "operating point" of 8 GHz processor speed, 2µs network latency, and 8Gbytes/s network bandwidth, some insight can be gained as to the relative importance of that sub-system through observation of the overall performance variation.

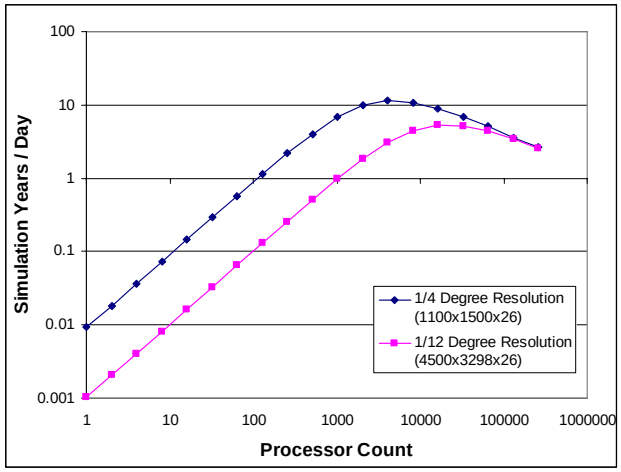


Figure 5: Number of HYCOM simulation days completed per hour on a possible future system

It can be seen that the processing speed is the most critical component of overall performance up to 512 processors. This indicates the HYCOM is, until this scale, computationally bound. With more processors, the performance of the network increases in importance. This is typical of applications that run in strong scaling mode. Latency is crucial, particularly at larger numbers of processors, as the interfaces between tiles shrink, leading to smaller message sizes. This is further borne out by examining the impact on performance by bandwidth (Figure 6, bottom). Past a certain scale, approximately 16K processors, as message sizes become smaller, available bandwidth plays less of a role in overall message passing performance.

6. Conclusions

We have presented a validated performance model of the HYCOM ocean simulation application. The model was developed from a thorough analysis of the computation and communication activities contained within the application, and is parameterized in terms of application characteristics, such as spatial grid size, computation cost per cell, and message passing activity, as well as system characteristics such as processing speed and communication performance. The model then proceeds through a validation stage, in which we demonstrate the accuracy of the model on a variety of parallel platforms and input decks.

Such models are vital tools to assess a priori the expected performance of a specific workload on a given parallel machine. It is possible to develop a qualitative analysis of expected application performance on parallel machines for which direct measurements are not possible, such as those machines which do not yet exist. It is also possible to assess which sub-system of the overall machine (e.g. processing speed, network latency, or network bandwidth) will have the most substantial impact on overall application performance. To this end, we have performed

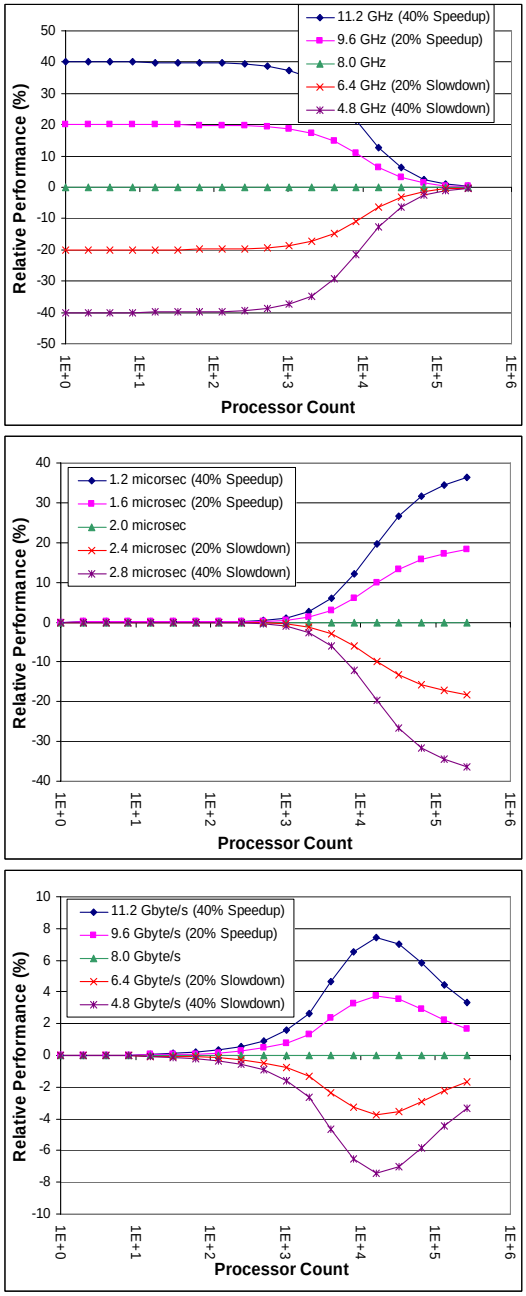


Figure 6: Performance sensitivity to processor speed (top), network latency (center), and network bandwidth (bottom) on spatial grid of 1/12 degree resolution

an analysis based on a possible machine architecture available within a five year timeframe and a workload based on a spatial grid encompassing the world's oceans at a 1/12 degree resolution.

This work is part of an ongoing activity to model a large scientific workload consisting of applications from both the Department of Defense and the Department of Energy. To date we have modeled applications including radiation transport, nondeterministic particle simulation, adaptive mesh hydro codes, and other elements of climate simulations. We have also used the models in a variety of ways including the comparison of large-scale systems and in the analysis of possible future systems and application optimizations. Within the DARPA HPCS program, we are using performance modeling to explore alternatives early in the system architecture design process.

Acknowledgements

This work is funded in part by the DARPA High Productivity Computing (HPCS) program. Los Alamos National Laboratory is operated by the University of California for the National Nuclear Security Administration of the United States Department of Energy.

References

[1] D.E. Keyes et. al. A science-based case for large-scale simulation, Vol. 2, Office of Science, Department of Energy, USA. Available from http://www.pnl.gov/scales

[2] R. Bleck, An oceanic general circulation model framed in hybrid isopycnic-cartesian coordinates, *Ocean Modeling*, *4*(1), 2002, 55-88.

[3] P.W. Jones, P.H. Worley, Y. Yoshida, J.B. White, J. Levesque, Practical performance portability in the parallel ocean program (POP), *Concurrency and Computation: Practice and Experience*, *17*(10), 2005, 1317-1327.

[4] D.J. Kerbyson, P.W. Jones, A performance model of the parallel ocean program, *Int. Jour. High Performance Computing Applications*, Special Issue 2005.

[5] D.J. Kerbyson, A. Hoisie, H.J. Wasserman, A performance comparison between the earth simulator and other terascale systems on a characteristic workload, *Concurrency and Computation: Practice and Experience*, *17*(10), 2005, 1219-1238.

[6] F. Petrini, D.J. Kerbyson, S. Pakin, The case of the missing supercomputer performance: achieving optimal performance on the 8,192 processors of ASCI Q, *Proc. IEEE/ACM Supercomputing*, Phoenix, AZ, 2003.

[7] S. Sun, J. Hansen, Climate simulations for 1951-2001 with a coupled atmosphere-ocean model, *Journal of Climate*, *16*(17), 2003, 2807-2826.

[8] W. Han, P.J. Webster, R. Lukas, P. Hacker, A. Hu, Impact of atmospheric intraseasonal variability in the Indian Ocean: low-frequency rectification in equatorial surface current and transport, *Journal of Physical Oceanography*, *34*(6), 2004, 1350-1372.

[9] C. Shaji, C. Wang, G.R. Halliwell, A.J. Wallcraft, Simulation of tropical pacific and atlantic oceans using a hybrid coordinate ocean model, *Ocean Modeling*, *9*(3), 2005, 253-282.

[10] HYCOM Consortium Web Page. Available from http://hycom.rsmas.miami.edu/